

# Dify knowledge documents and indexing

Use this family to create or replace knowledge documents, inspect their state, track indexing, control availability, download approved sources, or delete one document. All twelve operations use a workspace-scoped knowledge credential.

|          |                                                                       |
|----------|-----------------------------------------------------------------------|
| SECTION  | Connectors                                                            |
| TYPE     | Connector                                                             |
| PROTOCOL | AIP 1.0                                                               |
| SOURCE   | docs/connectors/dify/capabilities/knowledge-documents-and-indexing.md |

Document content, metadata, source links, and indexing errors can contain confidential data. Approval must identify the workspace, dataset, document, source content, processing change, and result handling.

## Choose an operation

| Operation suffix         | Method and route                                                      | Result                | Policy                                                   |
|--------------------------|-----------------------------------------------------------------------|-----------------------|----------------------------------------------------------|
| document.status.update   | PATCH /v1/datasets/{dataset_id}/documents/status/{action}             | JSON success          | Medium mutation; approval and key required               |
| document.text.create     | POST /v1/datasets/{dataset_id}/document/create-by-text                | Document and batch    | Medium mutation; approval and key required               |
| document.text.update     | POST /v1/datasets/{dataset_id}/documents/{document_id}/update-by-text | Document and batch    | Medium mutation; approval and key required               |
| document.file.create     | POST /v1/datasets/{dataset_id}/document/create-by-file                | Document and batch    | Medium mutation; approval and key required               |
| document.file.update     | POST /v1/datasets/{dataset_id}/documents/{document_id}/update-by-file | Document and batch    | Medium compatibility mutation; approval and key required |
| document.list            | GET /v1/datasets/{dataset_id}/documents                               | Paginated JSON        | Low-risk read; retry supported                           |
| document.download_zip    | POST /v1/datasets/{dataset_id}/documents/download-zip                 | Bounded ZIP           | Medium mutation; approval and key required               |
| document.indexing_status | GET /v1/datasets/{dataset_id}/documents/{batch}/indexing-status       | Status list           | Low-risk read; retry supported                           |
| document.download        | GET /v1/datasets/{dataset_id}/documents/{document_id}/download        | Signed URL descriptor | Low-risk read; retry supported                           |
| document.get             | GET /v1/datasets/{dataset_id}/documents/{document_id}                 | Document detail       | Low-risk read; retry supported                           |
| document.update          | PATCH /v1/datasets/{dataset_id}/documents/{document_id}               | Document and batch    | Medium canonical mutation; approval and key required     |
| document.delete          | DELETE /v1/datasets/{dataset_id}/documents/{document_id}              | Empty 204             | High mutation; approval and key required                 |

These Tool operations have knowledge scope and no app-mode projection. The capability template is:

```
TEXT
cap:dify:knowledge:<credential_id>:<operation>
```

Dataset and document route identifiers must belong to the selected workspace. The pinned provider requires UUIDs for those route values.

## List or inspect documents

`document.list` accepts `page`, `limit`, `keyword`, and `status` through Action `query`. Pagination defaults to page 1 and limit 20; the provider caps the limit at 100.

Status accepts `queuing`, `indexing`, `paused`, `error`, `available`, `disabled`, or `archived`. Keyword filters by document name.

```
JSON
{
  "dataset_id": "0190c42f-2d5a-7000-8000-000000000501",
  "query": {
    "page": 1,
    "limit": 20,
    "status": "available",
    "keyword": "refund"
  }
}
```

The response contains `data`, `has_more`, `limit`, `total`, and `page`. Records can expose source identity, name, token and word counts, indexing state, display state, metadata, hit count, document form, and summary state.

`document.get` accepts a `metadata` query value of `all`, `only`, or `without`. The default `all` includes document metadata. `only` returns `id`, `doc_type`, and `doc_metadata`; `without` omits the last two metadata fields.

## Create a text document

`document.text.create` requires `body.name` and `body.text`. It accepts processing rules, retrieval settings, model identifiers, and these document forms:

- `text_model` for ordinary chunks
- `hierarchical_model` for parent and child chunks
- `qa_model` for extracted question-and-answer pairs

```
JSON
{
  "dataset_id": "0190c42f-2d5a-7000-8000-000000000501",
  "body": {
    "name": "Refund policy",
    "text": "Approved customers may request a refund within the stated window.",
    "doc_form": "text_model",
    "doc_language": "English",
    "indexing_technique": "high_quality"
  }
}
```

The first document requires `indexing_technique` when the dataset has none. Later documents inherit the dataset setting. The default form is `text_model`, and the default processing language is English.

The provider stores the text as an uploaded source and returns `document` plus `batch`. Indexing continues asynchronously after that response.

## Update a text document

`document.text.update` accepts name, text, processing rules, form, language, and retrieval settings. When `text` is supplied, `name` is also required.

```
JSON
{
  "dataset_id": "0190c42f-2d5a-7000-8000-000000000501",
  "document_id": "0190c42f-2d5a-7000-8000-000000000502",
```

```

"body": {
  "name": "Refund policy, reviewed",
  "text": "Approved customers may request a refund within thirty days."
}
}

```

Content replacement creates a new source representation and re-triggers indexing. Preserve the returned batch rather than treating the HTTP response as completed indexing.

## Create or update from a file

All three multipart operations require Action `file` at the connector boundary:

```

JSON
{
  "dataset_id": "0190c42f-2d5a-7000-8000-000000000501",
  "file": {
    "filename": "refund-policy.pdf",
    "content_type": "application/pdf",
    "content_base64": "JVBERi0="
  },
  "body": {
    "doc_form": "text_model",
    "doc_language": "English",
    "indexing_technique": "high_quality"
  }
}

```

The connector sends the file plus serialized JSON in multipart data. It requires a safe basename and caps decoded content at 32 MiB. Provider file type, tenant quota, and configured upload limits can be stricter.

Use `document.file.create` for a new internal dataset document. File upload is not supported for an external dataset. The provider permits exactly one file and returns document plus batch.

Prefer canonical `document.update` for an existing document. The `document.file.update` operation uses the pinned provider's compatibility POST route. Both reach the same update helper and re-trigger indexing.

The provider PATCH route can update settings without replacement bytes, but the connector's multipart schema still requires `file`. This connector version does not expose settings-only PATCH input.

## Track indexing by batch

Every content create or replacement returns a `batch`. Call `document.indexing_status` with the same dataset and batch:

```

JSON
{
  "dataset_id": "0190c42f-2d5a-7000-8000-000000000501",
  "batch": "batch-0190c42f"
}

```

Normal progress moves through `waiting`, `parsing`, `cleaning`, `splitting`, `indexing`, and `completed`. A document can instead report `paused`, an error, or stopped evidence.

Each status record contains document ID, current phase, processing timestamps, error, and completed and total segment counts. A completed HTTP poll is not a terminal indexing state unless every intended document reports terminal evidence.

The connector has no provider indexing cancellation operation. Use status control only when its documented action matches the intended lifecycle change.

## Change document availability

`document.status.update` accepts `action` values `enable`, `disable`, `archive`, or `un_archive`. The body carries the bounded document set:

```
JSON
{
  "dataset_id": "0190c42f-2d5a-7000-8000-000000000501",
  "action": "disable",
  "body": {
    "document_ids": [
      "0190c42f-2d5a-7000-8000-000000000502"
    ]
  }
}
```

Dify checks dataset permission and model configuration before applying the batch action. It rejects invalid transitions, including changes that conflict with active indexing.

Success returns `result: success`. Confirm the resulting display state with a document read before releasing correlation evidence.

## Download approved sources

`document.download` returns a JSON object containing a signed `url` for the original uploaded file. It does not return the file bytes. Apply outbound URL, expiry, authorization, and content controls before following that URL.

`document.download_zip` accepts 1 to 100 UUID document IDs:

```
JSON
{
  "dataset_id": "0190c42f-2d5a-7000-8000-000000000501",
  "body": {
    "document_ids": [
      "0190c42f-2d5a-7000-8000-000000000502"
    ]
  }
}
```

Only approved uploaded-file documents enter the provider archive. The connector buffers and returns base64 ZIP data in its binary result envelope. Its configured response limit defaults to 8 MiB and permits 1 to 64 MiB.

The connector classifies ZIP generation as a mutation because it uses POST. It requires approval and a key and publishes no safe retry.

## Delete one document

`document.delete` permanently removes the document and all chunks. Dify rejects the request while the document is indexing or when it is archived.

Success returns empty status 204, represented as `null` in the generic connector result. There is no rollback or reconstruction contract.

## Apply mutation governance

All eight mutations require a non-blank Action idempotency key of at most 512 bytes. AIP retains its tenant-scoped reservation for 24 hours and revalidates the input hash on collision.

The key does not prove provider deduplication. These operations publish no automatic retry, transaction, reconciliation, indexing cancellation, or compensation support.

Treat text, files, metadata, signed URLs, ZIP bytes, processing rules, model choices, statuses, and errors as confidential PII-bearing data. Redact them from general logs.

## Recover safely

| Observation                                                    | Decision                                                                       |
|----------------------------------------------------------------|--------------------------------------------------------------------------------|
| UUID, action, query, body, or file is rejected before dispatch | Correct the bounded request without claiming a provider effect                 |
| Create or replacement returns a batch                          | Persist it and poll indexing status until terminal evidence                    |
| Mutation response is lost after dispatch                       | Read the document and indexing state before another mutation                   |
| Indexing pauses or errors                                      | Preserve the batch and error, repair the cause, then authorize the next action |
| Status change returns success                                  | Confirm the document display state through a read                              |
| Signed URL is missing or expired                               | Request a new descriptor after rechecking authorization                        |
| ZIP exceeds the response bound                                 | Reduce the approved document set; do not raise limits blindly                  |
| Delete is blocked by indexing or archive state                 | Resolve that state through an approved lifecycle action first                  |
| Returned content crosses a workspace or dataset boundary       | Stop traffic and treat the event as an isolation incident                      |

Preserve the Action ID, digest, approval, idempotency key, credential, dataset, document, batch, status, and bounded provider evidence needed to settle an ambiguous outcome.

## Related documentation

- [Dify capability index](#)
- [Dify knowledge bases, tags, and retrieval](#)
- [Dify connector configuration](#)
- [Authentication and credential scopes](#)
- [Errors and retry decisions](#)